

05_JS_Animation_Plane_w/Banner

jQuery & CSS Sprite Animation Explained in Under 5 Minutes

<http://addyosmani.com/blog/jquery-sprite-animation/>

Author: Addy Osmani – <http://addyosmani.com>

jQuery Spritely Example from AddyOsmani.com: <http://addyosmani.com/blog/jquery-sprite-animation/>

Duration: 01h 30m

Level: Intermediate

Project File Location: 158_Interactivity > Projects > 05_JQ_Animation > Ex_Files_JQ_Animation

Description:

In *jQuery & CSS Sprite Animation Explained in Under 5 Minutes*, author Addy Osmani builds a visually rich, animation using jQuery and the plugin Spritely. This tutorial covers preparing artwork in Photoshop, creating HTML and CSS layouts, and a sprite with the open-source JavaScript framework jQuery and the plugin Spritely. Tutorials on panning are also included. Exercise files accompany the tutorial.

Learning Objectives:

- Linking the HTML and CSS layouts
- Creating background images in Photoshop
- Creating a sprite
- Adding CSS rules for the animated layers
- Animating a sprite using a jQuery plugin called Spritely
- Utilizing the pan() function to move the background images in a particular direction
- Utilizing the sprite() function to work with multiple frames
- Adding more complex behaviors to animations

BEFORE YOU START:

Watch the 5 minute movie created by Addy Osmani explaining what a sprite is and how the layers animate in this tutorial.

Finished project: Folder > Final_Animation_Plane

– **Preview** first, then look inside folder to see file directory, structure and naming)

Tools:

jQuery – <http://www.jquery.com>

– Download the JavaScript files – Production (compressed) & Development (uncompressed)
JQ_Scripts Folder

Spritely Plugin - <http://www.spritely.net/>
spritely_download Folder

00: Introduction

—Previewing the project across browsers and devices – [View Final Project](#)

Ex_Files_JQ_Animation > Folder > Final_Animation_Plane > demo.html

—Using the exercise files

demo.html

js folder > jquery, Spritely plugin

images folder > city1.png, city2.png, sky.jpg, and planesprite4.png

- [Final Project Structure](#)

- [DIVS](#)

- #sky
- #city2
- #city1
- #plane1
- #content
- #leftcontent
- #rightcontent

01: Preparing Web Graphics

—Creating the background images & slices in Photoshop

1. In the Exercise Folder in the Art Templates folder, open up the file “animation_bg.psd” in Photoshop.
2. Turn off visibility for all layers except city1.
3. File > Export for Web & Devices; select PNG-24 and check transparency; then save into images folder (name: city1.png; size: 2220 x 312 px).
4. Repeat these steps again for the city2 layer. Then save after exporting for web.
5. Turn off visibility for the two city layers; only the sky background layer is visible; select the slice tool (under crop tool), and define a rectangle (2000 x 460 px).
6. Select File > Save for Web & Devices; zoom up; make sure slices is selected in the left hand column.
7. The settings is JPEG and 80% Quality. Then click Save and name sky.jpg.
8. Place the images folder into the myAnimation folder.
9. Close the animation_bg.psd file and Save.

02: Creating Project Folders

—Creating and linking HTML and CSS files

1. On the desktop, make a folder called myAnimation.
2. Inside this project folder create three folders - one called “js” to hold JS files, one called “css” and another called “images.”
3. In Dreamweaver, define your site with the root folder being myAnimation on the desktop.
4. Now create both HTML and CSS files. In Dreamweaver choose File > New (HTML Transitional) and save as index.html. Save.

5. Create File > New > CSS document. Save it as animation.css in css folder.
6. Now link the CSS file to the newly created index.html. If done properly you will see the animation.css is a related file and it will also show up in the head tag.

03: Marking up the HTML – index.html & creating the CSS styles

—Creating the divs for the background images, plane sprite, and content areas

1. In index.html, between the <body> tags, type the following HTML code:

```
<body>

<div id="sky"></div>
<div id="city1"></div>
<div id="city2"></div>
<div id="layer1"></div>
<div id="plane1"></div>

<br />

<div id="content">
    <div id="leftcontent"></div>
    <div id="rightcontent"></div>
</div>

</body>
```

2. Now Save All.
3. Now open up the animation.css file and we will begin to add styles. The background images are all placed in the background of their respective divs. All background elements and the plane are positioned “absolute” in relation to the browser window. They are not inside a container div in this tutorial. The width is 100% (of the browser window) so the layout stretched in width to the size of the browser window open. Also not the #content div clears the divs above it.

4. The styles are as follows:

```
body {
    background: #fff;
    margin:0 auto;
}
#sky {
    position:absolute;
    width:100%;
    height: 416px;
    background: transparent url(../images/sky.jpg) 0 0px repeat-x;
}
#city2 {
    position: absolute;
    width: 100%;
    height: 416px;
    background: transparent url(../images/city2.png) 0 130px repeat-x;
}
#city1 {
    position: absolute;
    width: 100%;
    height: 416px;
    background: transparent url(../images/city1.png) repeat-x 0 130px;
}
#plane1 {
    position: absolute;
    width: 730px;
    height: 167px;
    background: transparent url(../images/planesprite4.png) 0 50px no-repeat;
}
#content {
    width: 100%;
    position:relative;
    clear:both;
}
#leftcontent {
    width:70%;
}
#rightcontent {
    width:30%;
    background-color:blue;
}
```

5. Attach the jQuery (minimum) script to your index.html using the <script> tag; then create a new javascript file saved into the js folder called “animation.js”. Attache this script to index.html as well.
6. Save all.

04: Testing a simple example of the pan() function

—Creating a test of the pan() function

1. In your project folder, create another HTML page and save it as pan.html.
2. Next add this markup between the <body> tags:

```
<div id="city"> </div>
```

3. Then create a new CSS page called pan.css and save it into the CSS folder.
4. Write the following CSS styles:

```
body {

    background: #fff;

    margin:0 auto;

}
```

```

/*City Silhouette*/

#city {

    position: absolute;

    width: 99%;

    height: 416px;

    background: transparent url(city1.png) 0 130px repeat-x;

    border: 1px solid black;

}

```

5. Attach the jQuery file, then the spritely js file; then create a new JavaScript file called “pan.js” into the js folder and attach it with a <script> tag to the head. (You will have 3 JavaScript docs in the following order: 1) jQuery, 2) spritely, and 3) pan.js.)
6. Write the following code into the pan.js document:

```

$(document).ready(function() {

    $('#city').pan({fps: 30, speed: 2, dir: 'left'});

});

```



7. Save All and then preview the pan.html in the browser. Play around with the fps, speed, and direction parameters.

05: Creating a sprite with numerous key frames

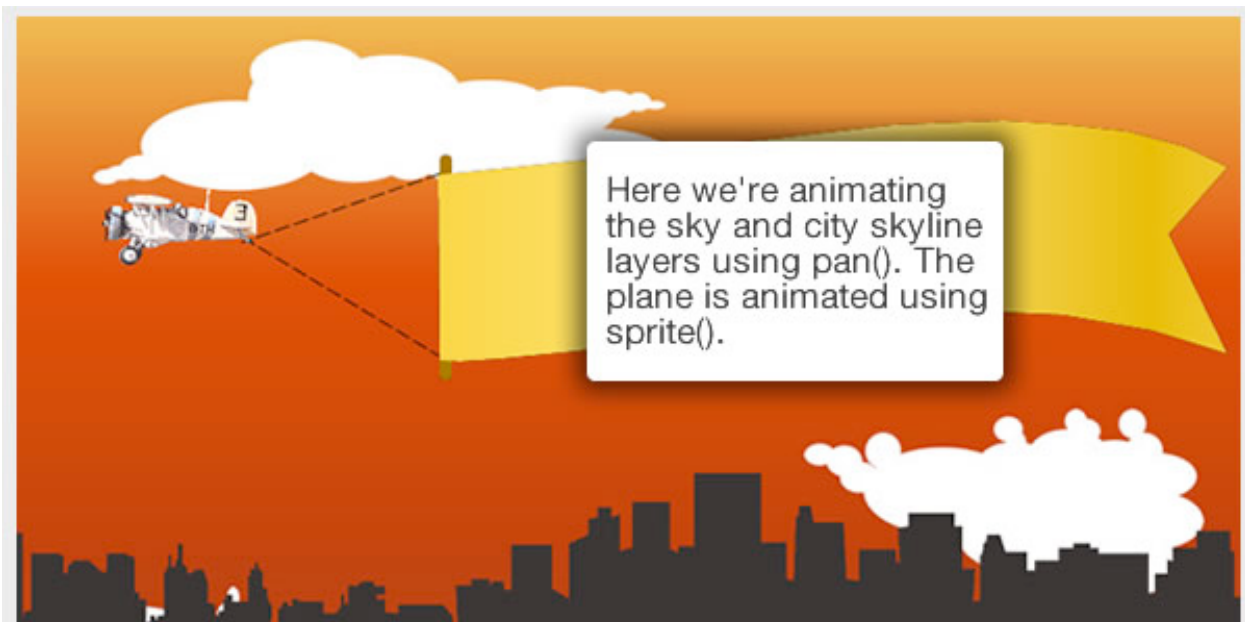
—Outputting your frames

1. Output your frames as transparent PNG files to a directory on your device. Since animation can sometimes be a memory intensive task, minimize as much as possible. This can be achieved by outputting 1 frame for every 5 – 15 in your original animation (if you used an animation program like Maya, Flash, AfterEffects, Keynote, etc). Don't worry! The Spritely plugin supports custom frame-rates so you can still make your final animation look relatively smooth.
2. Use a Sprite Generation tool such as CSSSprite.es or CSSSprites.com to stitch the your frames together as a single image file. Try to keep the final file size down to within a few megabytes (recommended 2 MB or less; 2MB would be the high end of the file size).
 - **CSS Sprites:** <http://css-sprit.es/>
 - **CSS Sprites Generator:** <http://csssprites.com/>
3. Next animate your content Spritely by going onto the next chapter using the Spritely plugin and jQuery using the `sprite()`.

06: Using Spritely to animate a sprite

—Writing the jQuery code using the Spritely plugin

1. In the index.html, attach the 3 JavaScript files – jQuery, Spritely, and animate.js using the `<script>` tag. Make sure the JavaScript files are placed in the above order.



1. In the animation.js, write the following functions to animate the background images and sprite (airplane), as follows:

```
1. (function($) {
2.     $(document).ready(function() {
3.         /* City 1 */
4.         $('#city1').pan({fps: 30, speed: 1, dir: 'right'});

5.         /* City 2 - This gives the animation a depth of perspective
6.         as one city silhouette is slightly away from the other*/
7.         $('#city2').pan({fps: 30, speed: 2, dir: 'right'});

8.         /*The clouds and sky*/
9.         $('#sky').pan({fps: 30, speed: 0.5, dir: 'right'});

10.        /*The Plane animation*/
11.        var stage_left = (($('body').width() - 866) / 2);
12.        var stage_top = 30;

13.        /* Spritely's sprite function allows you to define
14.        more complex animations which contain more than 1
15.        frame. You can also use the spRandom feature to define
16.        random behaviour.
17.        */
18.        $('#plane1').sprite({fps: 8, no_of_frames: 14})
19.            .spRandom({
20.                top: 40,
21.                left: stage_left + 20,
22.                right: 400,
23.                bottom: 140,
24.                speed: 3500,
25.                pause: 1000
26.            });
27.    });
28. })(jQuery);
```

2. Save All and test in browser. Now go back into animate.js and “play” around with parameters of the animation.

Vocabulary

- pan()
- sprite()
- fps
- speed
- dir
- spRandom

Due date: 2 weeks